

## LA-UR-19-30566

Approved for public release; distribution is unlimited.

Title: Combining Observational and Computational Uncertainty in Calibration Experiments

Author(s): Foster, Robert Christian  
Gattiker, James R.  
Weaver, Brian Phillip

Intended for: Report

Issued: 2019-10-17

---

**Disclaimer:**

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# Combining Observational and Computational Uncertainty in Calibration Experiments

Robert C. Foster, Jim Gattiker, Brian Weaver

Los Alamos National Laboratory

October 16, 2019

## Abstract

Traditional computer experiments using emulators have treat the emulator output as containing no error; however, this is not always feasible, as there exist computational codes which produce stochastic output, and future computations may require a balance between the time or resources to run the code and the accuracy of the output. This paper proposes a framework for incorporating both computational uncertainty and observational uncertainty into an analysis using a Gaussian process regression and calibration, and provides an example using simulated output from the one-dimensional diffusion equation. The particular values of observational and computation uncertainty which minimize a cost function are obtained.

**Keywords**— Computer experiments, Gaussian process, Experimental design, Approximate computing

# 1 Introduction

In engineering and the sciences, experiments are often costly in terms of time and money. As such, it is common to combine data from physical experiments with data from a computer simulator of the system under study. These are typically combined with a Gaussian process, as in Kennedy and O’Hagan (2001). Such simulators are often stochastic, producing random output as a function of the inputs. Even in simulators that are considered deterministic, the experimenter will likely have control over aspects of the code which can be manipulated to increase or decrease the precision of the output.

This paper proposes combining observations from physical systems with simulated data points from the computer emulators and incorporates both observational error on the physical system and calculation error from the computer model. This can be used to investigate the outcome of trading off between accuracy in observed data points and accuracy in simulated data points, both in terms of statistical measures of the uncertainty in the final analysis and measures of the actual costs in time and money.

Section 2 describes a framework for computer experiments which includes both sources of uncertainty. Section 3 proposes a general iterative method and describes terminology for obtaining contours in the space of experimental designs. Section 4 shows an application of the previously described framework and terminology to a simple Gaussian process regression emulator representing the outcome of the one-dimensional diffusion equation.

## 2 Ambiguous Observation and Approximate Computation

Suppose that an experiment measures the outcome of a physical system under the influence of one or more factors. Define  $\theta$ , which can be a scalar or vector, as the chosen level or levels of the factors. It is typical to directly manipulate  $\theta$  and collect observed data points of the system. Define these observed data points as

$$\begin{aligned} y^{obs} &= \eta(\theta) + \epsilon_{obs} \\ \epsilon_{obs} &\sim N(0, \sigma_{obs}^2) \end{aligned} \tag{1}$$

where  $\eta(\theta)$  is the physical response as a function of the factor level(s)  $\theta$ . These have observational error, which is commonly modeled as normally distributed with mean 0 and variance  $\sigma_{obs}^2$ , and which may be a sum of multiple independent sources of error.

If time and money were infinite, then an experiment would be conducted entirely with  $y^{obs}$  and analyzed in a traditional fashion, such as with an analysis of variance. In the sciences, however, such observations can be expensive and time-consuming to collect, and so observations of the physical system are often paired with simulated observations generated from emulators at different levels of the given factors.

Realistic emulators of physical systems commonly operate by solving collections of partial differential equations which are believed to accurately model the physical system in question. For the state of the system at location  $x$  and at time  $t$ , define

$$u_{x,t} = u(x, t) \quad (2)$$

where  $u_{x,t}$  is taken to be the “true” state of the emulator (not the true physical system), absent all sources of uncertainty. Note that  $u_{x,t}$  can be either a scalar or vector.

These emulators can also be time consuming to run, and are often stochastic, producing output which can be modeled as a random variable as a function of the inputs. The experimenter generally has a degree of control over the degree of stochasticity of the output, however, and can reduce the time and accuracy required at the cost of precision of the solution. For example, methods which operate by stepping forward discretely in time can reach the solution faster by taking larger time steps, but at the cost of a worse approximation to the “true” solution  $u_{x,t}$ . This can be seen as a form of adding additional noise into the solver. The resulting output is often treated as deterministic, but it should be seen as random variable which is a function of the step size.

Suppose one such solver is used with time step  $h$ . The state of the system  $u_{x,t+h}$  is calculated at each successive step as a function  $f(u_{x,t})$  of the current state of the system at a given location. The noise introduced by the time discretization (or linear discretization error, or any other source of computational uncertainty) is additional error that can be modeled as an additive error to each step:

$$u_{x,t+h}^* = f(u_{x,t}^*) + \delta_{x,t} \quad (3)$$

Once again,  $u_{x,t}^*$  and  $\delta_{x,t}$  may be either scalars or vectors. The initial values  $u_{x,0}$  are assumed to be known with no uncertainty, though if this is not the case the initial values can be seen as including noise  $\delta_{x,0}$  with no additional change to the framework. Note that the states are now  $u_{x,t}^*$  instead of  $u_{x,t}$  - the notation  $u_{x,t}^*$  is taken to be  $u_{x,t}$  with uncertainty. Though any error distribution may be used in practice, for this paper the  $\delta_{x,t}$  are taken to be independent draws from a normal distribution with mean 0 and standard deviation  $\sigma_{comp}$ .

$$\delta_{x,t} \sim N(0, \sigma_{comp}) \quad (4)$$

The standard deviation  $\sigma_{comp}$  represents the control the experimenter has over the stochasticity of the code. By increasing or decreasing  $\sigma_{comp}$ , for example by increasing or decreasing the time step  $h$ , the experimenter can make the results more or less accurate as desired.

Suppose the system is run until a stopping time  $T$ . The final system state, with computational uncertainty, is then modeled as

$$u_{x,T}^* = u_{x,T} + \epsilon_x$$

The observation is assumed to have a single additive error  $\epsilon_x$ . Again, though any error distribution may be used, for this paper the errors are taken to be normally distributed with mean 0 and standard deviation  $\sigma_{calc}$ .

$$\epsilon_x \sim N(0, \sigma_{calc})$$

Note that  $\sigma_{comp}$  and  $\sigma_{calc}$  are not equivalent - because the process applies the (possibly nonlinear) solver to noisy observations repeatedly, compounding the noise at each step, the stepwise errors may interact in unexpected ways, and the error on the final step may have either larger or smaller variance than the stepwise additive noise.

In general,  $\sigma_{calc}$  is the quantity of interest. More complex codes outside of the examples of this paper will generally have multiple inputs to control the stochasticity of the code, and the effect of manipulating the inputs will be more than simply an additive error with a single variance, but the cumulative effects of all the errors will still produce a single error for each output. Even when  $\sigma_{comp}$  is an unreasonable assumption, or the specific amount of stepwise error introduced is difficult to determine in practice, a  $\sigma_{calc}$  remains and should be estimated.

The simulated data point is then generally some function of the final state of the system. This should serve as an estimate of  $\eta(\theta)$

$$y^{sim} = h(u_{x,T}^*) = h(u_{x,T} + \epsilon_x) = \widehat{\eta(\theta)}$$

The simplest case, which is used in this paper, is to simply take  $h(u_{x,T}^*) = u_{x,T}^*$ .

$$\begin{aligned} y^{sim} &= \widehat{\eta(\theta)} = u_{x,T} + \epsilon_x \\ \epsilon_x &\sim N(0, \sigma_{calc}) \end{aligned} \quad (5)$$

Hence, the emulator output  $y_{sim} = u_{x,T}^*$  may be seen as an estimate of a “true” state  $u_{x,T}$  and an error state  $\epsilon_x$ . Then combining  $y^{obs}$  points as in Equation (1)

with  $y^{sim}$  points as in Equation (5) allows for a computer experiment in which both observational and computational uncertainty are incorporated.

## 3 Costs and Boundaries

### 3.1 Iterative Algorithm

Given a set of observed data points  $y_i^{obs}$  for  $i = 1, \dots, n$  as in equation (1) and simulated data points  $y_i^{sim}$  for  $i = 1, \dots, m$  as in equation (5), these points are typically combined with a Gaussian process in order to estimate the response as a function of the inputs, as in Kennedy and O’Hagan (2001) and Higdon et al. (2004). The presence of error in the emulator output has thus far been ignored, but it plays an important role in certain types of codes, such as in Baker et al. (2015).

Ideally, the experimenter would like to choose values of  $\sigma_{obs}$  and level of computational error leading to a desired  $\sigma_{calc}$  before the experiment is conducted that meet some threshold of uncertainty on the final output, but which still fit within the experimental budget. Decreasing  $\sigma_{obs}$  is, in many experiments in the sciences, not a simple task - construction of new facilities that allow for increased precision of measurements can cost millions of dollars or more. Decreasing the level of computational error, by for example decreasing the time step, can lead to codes which run for weeks or months at a time to generate a single point.

This leads to the obvious dilemma: how does one estimate the properties of an analysis which uses observational data, before collecting the data? In the case of the physical sciences, there may be data from preexisting experiments which can be used (possibly with modification) to provide an approximation to the unknown future experimental data. The emulator also provides a reasonable solution: simulated data points from the emulator can be used as “stand-in” points for the observed data points. In particular, high-accuracy emulator data points in which computational error is negligible should be used to avoid confounding computational uncertainty with observational uncertainty. It is likely that the emulator still contains bias; however, so long as the bias is not too significant the resulting analysis is unlikely to be largely affected. If bias is a concern, the computer output may be modified by some quantity to fit a standard discrepancy term in the analysis. Using either prior experimental data or high-accuracy emulator data points represents a reasonable compromise between repeatedly performing physical experiments (which, if it were possible, computer emulators would not be used) and simply guessing. The experimenter can then repeatedly perform a pseudo-experiment using the stand-in  $y^{obs}$  to observe the resulting quantity of interest.

Suppose the quantity of interest is an  $\alpha\%$  interval length from the posterior distribution for some unknown factor level  $\theta$ . This is the classic calibration problem. The following iterative method is proposed to determine optimal levels of  $\sigma_{obs}$  and computational error, leading to a desired  $\sigma_{calc}$ , for determining the posterior interval length.

1. Determine the experimental design, including the locations in parameter space where  $y^{obs}$  and  $y^{sim}$  data will be collected. For the  $y^{obs}$ , either obtain pre-existing experimental data or, using the computer emulator, conduct high-accuracy simulations at the chosen locations in parameter space.
2. Choose a given level of observational error  $\sigma_{obs}$  and level of computational error  $\sigma_{comp}$ , simulate  $y^{sim}$  data using the emulator with chosen level, conduct the analysis as planned with chosen  $\sigma_{obs}$ , and obtain  $\hat{\sigma}_{calc}$  and the posterior interval length for  $\theta$ . This is one pseudo-experiment.
3. Fit a response surface to the set of posterior interval lengths for  $\theta$ . The specific method is not important - it can be a polynomial regression, a Gaussian process, or something else - so long as predictive intervals around the surface can be obtained.
4. Estimate or obtain the uncertainty surrounding each point on the response surface, and determine the new points  $\sigma_{obs}$  and  $\sigma_{comp}$  which meets some objective of uncertainty reduction, potentially involving cost.
5. Repeat steps 2-4 until the experimenter is satisfied that the set of  $\sigma_{obs}$  and  $\hat{\sigma}_{calc}$  values has been found which meets some predetermined criteria.

Though the posterior interval length for a parameter  $\theta$  is used in this paper, the above process is not dependent on it, and can be used for any other reasonable posterior quantity of interest.

This iterative procedure faces two main obstacles: first, that variance around the estimated response surface of step 3 will not be constant, as there will be regions wherein the resulting  $\alpha\%$  confidence interval length will vary depending on whether data is generated randomly which is sufficient to obtain a good estimate of  $\theta$  or not, and second, that a method of choosing a new point on the response surface based upon the the current response surface, as in step 4, must be developed. Both of these questions will have solutions which vary depending on the specific properties of the problem. There are two methods of obtaining a response surface that will likely be used the most: a Gaussian process or a polynomial response surface. For Gaussian process models, the issue of determining a

point which maximizes some objective function, or points which form some contour, have been explored in Ranjan et al. (2008), and Bect et al. (2012), Chevalier et al. (2014), and Marques et al. (2018). Of these, Bect et al. (2012) provides the most complete description of various strategies for point selection. Unfortunately, Gaussian process models traditionally model the variance as constant, which is known to be untrue for this particular problem.

The second option, a polynomial response surface is typically fit using a least squares method, and this presents the natural solution of using weighted least squares in order to fit a model with a non-constant variance. The weights are typically taken as proportional to the variance at each location. The primary difficulty in the use of this method is determining the appropriate variance at each location based off of a single point. It should be noted, however, that the variance is likely to be largest in the areas of most rapid change, and so estimated derivatives may provide some clues as to the appropriate weights. There may also exist prior information which may be used to determine the variance, or it may be possible to use the model itself to estimate an error weight.

### 3.2 Notation for Optimization

The iterative procedure of Section 3.1 demands a precise notation for use in determining a specific goal for optimization and incorporating any other quantities of interest into the algorithm.

Define  $L_\alpha(\sigma_{obs}, \hat{\sigma}_{calc})$  as the resulting  $\alpha\%$  interval length from a single pseudo-experiment as described in step 2. Due to the stochastic nature of both the observed and simulated data points, this is necessarily a random quantity. The choice is to use  $\hat{\sigma}_{calc}$  rather than  $\sigma_{comp}$  because there should be a strong, if still stochastic, relationship between  $\sigma_{comp}$  and  $\hat{\sigma}_{calc}$ , and because costs associated with a certain level of uncertainty may more accurately be expressed on the resulting output uncertainty, not on the input uncertainty. If there exists a good reason to use  $\sigma_{comp}$ , however, it may be used with no other changes to the subsequent procedure.

For each  $(\sigma_{obs}, \hat{\sigma}_{calc})$  pair, define

$$G_K(\sigma_{obs}, \hat{\sigma}_{calc}) = P(L_\alpha(\sigma_{obs}, \hat{\sigma}_{calc}) < K)$$

where  $K$  is a target interval length, chosen a priori, and  $P$  is a standard probability measure. The specific value of  $K$  will depend on both the goals of the experiment and the experimenter's level of comfort with different levels of uncertainty, but assume it is constant and known. A reasonable goal, then, is to find a region in the space of  $(\sigma_{obs}, \hat{\sigma}_{calc})$  such that

$$B_{p,K} = \{(\sigma_{obs}, \hat{\sigma}_{calc}) \mid G_K(\sigma_{obs}, \hat{\sigma}_{calc}) \geq p\}$$

for some given probability  $p$  and target interval length  $K$ . Presumably,  $p$  will be chosen as large value, though it will likely depend on the problem at hand. Assume that it is fixed and chosen a priori.  $B_{p,K}$  then gives the set of  $(\sigma_{obs}, \hat{\sigma}_{calc})$  values such that the probability  $G_K$  of the resulting  $\alpha\%$  interval length  $L_\alpha$  being less than  $K$  is greater than or equal to  $p$ .

Lastly, define  $C(\sigma_{obs}, \hat{\sigma}_{calc})$  as the (usually nonlinear) cost of using  $\sigma_{obs}$  and the chosen level of computational error for the analysis, however cost is defined in the context of the problem. Then taking  $C(B_{p,K})$  produces a cost for each set of  $(\sigma_{obs}, \hat{\sigma}_{calc})$  in the set of values that give the target interval length. Using this, the experimenter can take budgetary constraints into the decision-making process. It may be, for example that identical results can be obtained from simply investing more time into the approximate computation to decrease  $\hat{\sigma}_{calc}$  rather than construction of an expensive new facility to decrease  $\sigma_{obs}$ .

Suppose that  $\alpha = 0.95$ ,  $K = 2$ , and  $p = 0.8$ , indicating that a 95% confidence interval is desired for the quantity of interest and the goal is to determine the boundary of the region such that the probability of the interval length being less than 2 is 0.80. In notation, the goal of the iterative procedure as described in Section 3.1 would be to determine  $B_{0.8,2}$ , which is the set of  $(\sigma_{obs}, \hat{\sigma}_{calc})$  at which the probability  $G_2(\sigma_{obs}, \hat{\sigma}_{calc})$  that the interval length  $L_{0.95}(\sigma_{obs}, \hat{\sigma}_{calc})$  is less than 2 is greater than or equal to 0.8. This will be a contour in the space of  $(\sigma_{obs}, \hat{\sigma}_{calc})$  values. Then  $C(B_{2,0.8})$  would produce a set of costs associated with each point in the region, from which the optimal value could be chosen.

## 4 Example: Diffusion Equation

The one-dimensional diffusion equation is a partial differential equation of the form

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} \quad (6)$$

This equation is numerically stable and can be solved by a number of computational methods, given initial and boundary conditions. Define  $u_{x,t} = u(x, t)$  as in equation (2). For all instances of the system in this paper, the initial condition is given by the function  $u_{x,0} = 1 - x^2$  for  $x$  in the range  $[-1, 1]$ . Dirichlet boundary conditions are  $u_{-1,t} = u_{1,t} = 0$  for all  $t$ . A total of  $N_x = 100$  equidistant points are used between  $-1$  and  $1$  using this initial condition, and a total of  $N_t = 100$  time steps are taken with step size  $h = 0.05$  up to a maximum time of  $T = 5.00$ . A backward Euler method coded in the R programming language

from the modified code of Gardner (2018) was used to find the solution  $u_{x,t}$  given for the partial differential in Equation (6). The state of the system in the range  $[-1, 1]$  is shown for three time steps of a run with diffusion constant  $D = 1$  below in Figure 1.

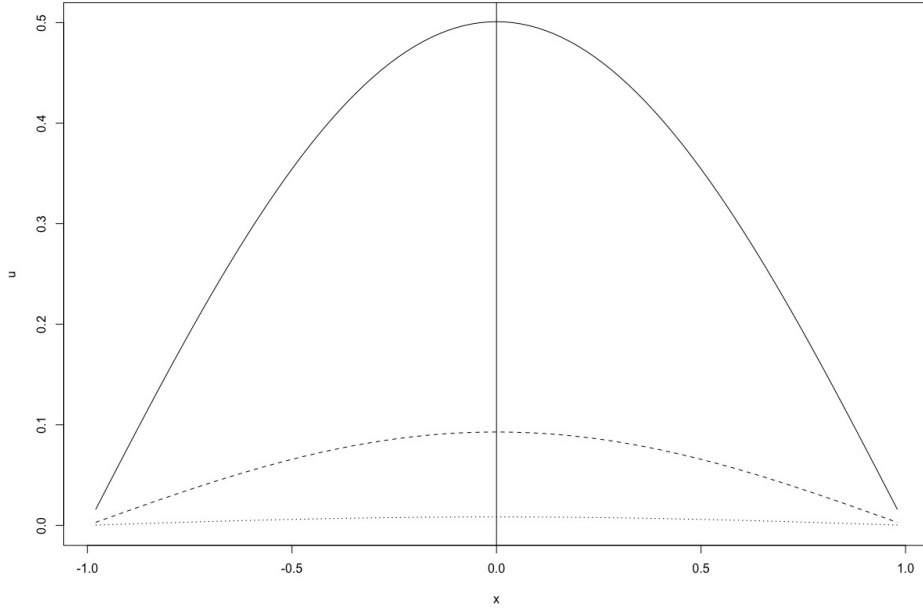


Figure 1: A single solution of the diffusion equation with diffusion constant  $D = 1$  is shown at time  $t = 0.05, 2.50$ , and  $5.00$ . The point  $u_{0,5.00}$  is taken as the observation for a solution of the system for a given diffusion constant - the value of  $u_{0,t}$  is indicated by the vertical line.

A key question is whether the diffusion constant  $D$  for a given system can be identified from the output of a single run. For the set of initial conditions used in this paper, the answer is yes - and in fact, only a single point is needed to make the identification. Though any point can reasonably be used, take  $u_{0,5.00}$  as the observation from a single run of a single system. The value of this point follows a monotonic negative exponential model as a function of the diffusion constant  $D$ , as shown below in Figure 2.

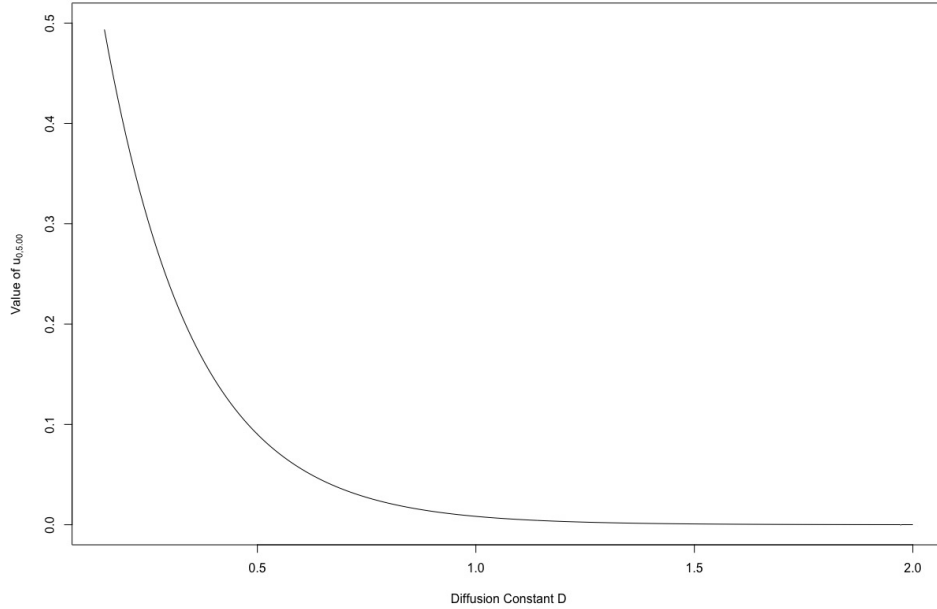


Figure 2: The value of  $u_{0,5.00}$  after the final time step is shown as a function of the diffusion constant  $D$ . The plot clearly shows a monotonic, nonlinear, negative exponential relationship between the two. Given a positive value for  $u_{0,5.00}$ , it is possible to precisely identify a value of  $D$  which produces that observation.

If  $u_{0,5.00}$  is known with perfect accuracy, then the monotonicity of the response in Figure 2 allows the corresponding value of  $D$  to also be identified with perfect accuracy. Suppose, however, that the code is stochastic and that for reasons of time or money, accuracy of computation must be sacrificed. A range of values is then possible for  $D$  which, if combined with noisy physical observations, allows calibration to proceed with both observational and computational noise.

## 4.1 Gaussian Process Calibration Model

Following Section 2, suppose that an experiment is to be performed for which a run of the diffusion equation serves as an emulator for a physical system given diffusion constant  $D$ . As in the previous section, the goal is to determine  $D$  from  $u_{0,5.00}$ . Let  $y^{sim}$  and  $y^{obs}$  be as defined in Section 2, where each data point is the observation  $u_{0,5.00}$  from either the computer emulator or the observed phys-

ical system, respectively. Though the step size  $h$  has been discussed a form of computation error, in one-dimensional systems such as the diffusion equation the effect of this added noise is so small as to be imperceptible in most cases. Instead, random noise is artificially added at each iteration of the solver as in equation (3). The noise was normally distributed with mean 0 and standard deviation  $\sigma_{comp}$ , as in equation (4). By directly controlling the value of  $\sigma_{comp}$ , the control that the experimenter has over the stochasticity of the code is approximated.

The model for the simulated points, with approximate computation noise  $\sigma_{calc}$  and observational noise  $\sigma_{obs}$ , can be written as

$$y_i^{sim} \sim GP(\mathbf{0}_m, K(D_i, D_j) + \sigma_{calc}^2 I_{m \times m}) \quad (7)$$

where  $\sigma_{calc}$  is as in Section 2 and the correlation function  $K(D_1, D_2)$  is defined as

$$K(D_i, D_j) = \gamma^2 \exp \left[ -\frac{(D_i - D_j)^2}{2\rho^2} \right] \quad (8)$$

The parameters  $\gamma$ ,  $\rho$ , and  $\sigma_{calc}$  are unknown and must be estimated.

In lieu of physical observations, high-precision points were used from the emulator, as described in Section 3. These were calculated as results from the emulator, but using  $\sigma_{comp} = 0$ . These points were identical for all data sets, as in a real experiment they would likely be time-consuming to compute, and would not be recreated for every iteration of the procedure described in Section 3.1.

Suppose there are  $n$  observed data points  $y_i^{obs}$  for  $i = 1, \dots, n$ . The  $y_i^{obs}$  are assumed to have a known  $\theta_i$  as the corresponding diffusion constant to be calibrated. Then the observed points also follow a Gaussian process

$$y_i^{obs} \sim GP(\mathbf{0}_n, K(\theta_i, \theta_j) + \sigma_{obs}^2 I_{n \times n}) \quad (9)$$

where  $K(\theta_i, \theta_j)$  is as in Equation (8). The standard deviation of the observation error  $\sigma_{obs}$  is assumed to be known, as the experimenter may have domain specific knowledge of the uncertainty surrounding observations of the system.

The same joint Gaussian process is assumed for both the simulated and observed data points in equations (7) and (9). The main difference between the two sets of points is that for each  $y_i^{sim}$  the diffusion constant  $D_i$  is known but the standard deviation  $\sigma_{calc}$  of the noise on the observations must be estimated, while for  $y_i^{obs}$  the standard deviation  $\sigma_{obs}$  on the noise is known but the diffusion constant  $\theta_i$  must be estimated. The goal is to fit a joint Gaussian Process regression to the  $y_i^{sim}$  and the  $y_i^{obs}$ , estimating  $\gamma$ ,  $\rho$ , and  $\sigma_{calc}$  while simultaneously calibrating the  $\theta_i$  corresponding to each observation.

## 4.2 Data Simulation

The experimenter would like to know the effect of manipulating each amount of noise on the resulting analysis in order to determine acceptable values of  $\sigma_{obs}$  and  $\sigma_{calc}$ . To that end, a total of 28 equally spaced values between  $10^{-4}$  and  $10^{-1}$  were chosen for both  $\sigma_{obs}$  and  $\sigma_{comp}$ . These values were used to generate random data from the model described in Section 4.1. A Gaussian process was fit to these data, including calibration for unknown  $\theta_i$ . For each pair of  $(\sigma_{comp}, \sigma_{obs})$  values, a total of 100 data sets were simulated, for a total of 78400 data sets.

For each data set, the diffusion constants chosen for simulation were  $D_{Sim} = \{0.125, 0.575, 1.025, 1.475, 1.925\}$ . At each value of  $D_{Sim}$ , a total of  $N_{rep} = 5$  runs of the solver given in Section 4 were performed, with added computational error as described in equations (3) and (4) in Section 2.

Three values of  $D$  were chosen for calibration, with a total of 1 data point per value:  $D_{Obs} = \{0.3125, 0.5600, 1.7500\}$ . For each data point, the point  $y^{obs}$  was calculated, and assumed to be a draw from a  $N(0, \sigma_{obs}^2)$  distribution. Following Section 2, these observations were calculated from the solver with no added error.

This gives a total of  $m = 25$  simulated data points  $y_i^{sim}$  and  $n = 3$  observed data points  $y_i^{obs}$ . For each new data set, the  $y_i^{sim}$  are generated anew. The  $y_i^{obs}$  are identical for all data sets. An example data set and corresponding GP regression is shown below in Figure 3.

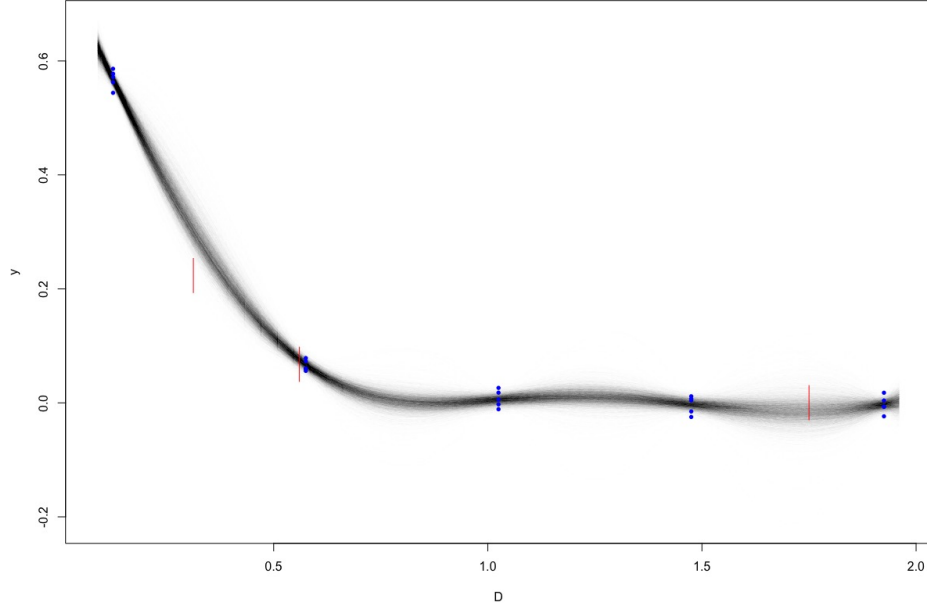


Figure 3: The output from a single run of the combined estimation and calibration for the diffusion equation using a Gaussian process is shown. The red lines represents the observed points  $y_i^{obs} \pm 3\sigma_{obs}$ , and their location along the horizontal axis is given by the mean posterior calibrated value of  $D$ . The blue points are simulated data points  $y_i^{sim}$  at known values of  $D$ , each of which used five points. The shaded black lines represent draws from the Gaussian process regression.

The particular values of  $D_{New}$  were chosen to be “moderate”, “difficult”, and “very difficult” to calibrate, respectively. So long as the noise is not large in both simulations and observations, the model should realistically be able to calibrate  $D_{Obs} = 0.3125$ , as it lies within a region of rapid change of  $u_{0,5.00}$  as a function of the diffusion constant, even with added uncertainty. Conversely, except in cases of very small noise in both simulations and observations, the model should be unable to calibrate  $D_{Obs} = 1.7500$ , as it lies in a region of very slow change of  $u_{0,5.00}$  as a function of the diffusion constant, so that with added uncertainty from the model virtually any diffusion constant from roughly  $D = 0.75$  to the maximum of  $D = 2$  is possible.

### 4.3 Estimation

A Bayesian MCMC approach was used to estimate the model. Using priors  $\rho \sim \text{Inverse Gamma}(5, 5)$ ,  $\gamma \sim N(0, 1)$ ,  $\theta_i \sim \text{Uniform}(0, 2)$ , and  $\sigma_{calc} \sim \text{Uniform}(0, 1)$ , a total of 25000 draws were taken after 5000 burn-in draws for each of four chains. Initial values were  $\rho_{(0)}^* = 0.8$ ,  $\gamma_{(0)}^* = 0.65$ ,  $\sigma_{comp,0}^0 = \sigma_{calc}$ , and  $\theta_{0,i}^* = D_{sim,i}$  for all four chains. essentially using “true” values used to generate the data where possible, and reasonable estimates where not possible. The STAN programming language was used to perform the MCMC. Though it is not possible to check all 78400 sets of chains, a random sample from the data sets showed good convergence in all chains.

Quantities measured include the mean, standard deviation, and 95% central interval lengths for the posteriors of each of the parameters  $\gamma, \rho$ , and  $\sigma_{calc}$ , and for each of the three calibrated diffusion constants  $\theta_1, \theta_2$ , and  $\theta_3$ . Intervals were obtained simply by taking quantiles from the combined posterior chains past the burn-in period. Two example calibration distributions for  $\theta_1$  and  $\theta_3$  in a given run of the solver are shown below in Figure 4.

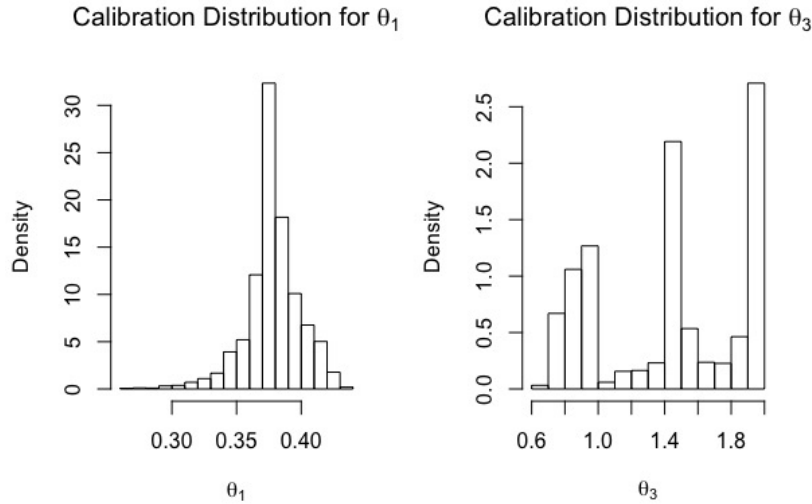


Figure 4: For a calibrated run of the diffusion equation with  $\sigma_{obs} = 10^{-4}$  and  $\sigma_{comp} = 10^{-4}$ , the calibration posterior distributions for  $\theta_1$  and  $\theta_3$  are shown. The diffusion equation constant  $\theta_1$  is well identified, but the posterior distribution for  $\theta_3$  is multimodal, indicating a number of values which would fit equally well.

## 4.4 Confidence Interval Lengths

Consider the length of the 95% interval for  $\theta_1$ , the unknown diffusion constant corresponding to the  $y_i^{obs}$  made with  $D_{obs} = 0.3125$ , and which the model should be able to calibrate except in the presence of large noise on both simulations and observations. In terms of the notation of Section 3.2, this is  $L_{0.95}(\sigma_{obs}, \hat{\sigma}_{calc})$ . Plots of the mean length of the confidence interval over the 100 total simulations at each  $(\sigma_{obs}, \hat{\sigma}_{calc})$  pair and for each are shown below in Figure 5.

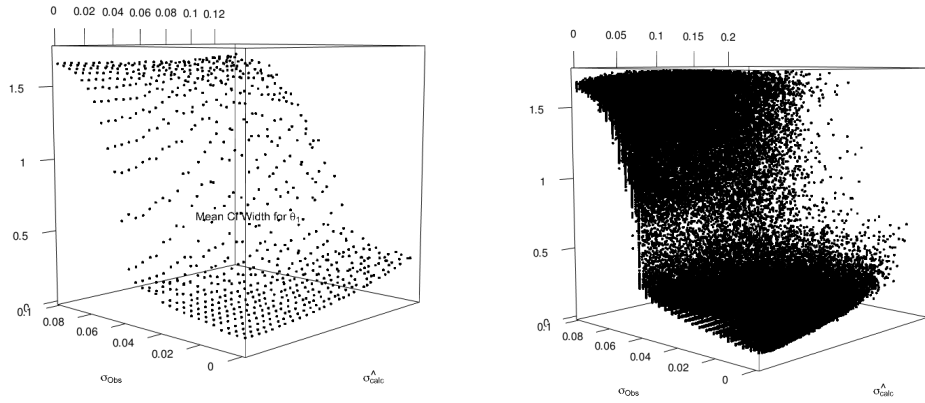


Figure 5: On the left, a plot showing the mean 95% interval width for the calibrated value of  $\theta_1$ , as a function of both  $\sigma_{obs}$  and mean  $\hat{\sigma}_{calc}$  for a given  $\sigma_{comp}$ . On the right, a plot of the 95% interval width for each of the 78,400 runs for the calibrated value of  $\theta_1$  as a function of  $\sigma_{obs}$  and  $\hat{\sigma}_{calc}$ . Notice the sigmoidal shape - as the uncertainty increases for either source, the interval length grows as the model is increasingly unable to identify an accurate value. For the largest values of each uncertainty, the posterior is multimodal over the range of considered diffusion constants.

The shape is essentially sigmoidal. For large value of both  $\sigma_{obs}$  and  $\hat{\sigma}_{calc}$  (induced by a specified  $\sigma_{comp}$ ), the confidence interval width is wide. For small values, the confidence interval width is small. For intermediate values, there is a clear decreasing average interval width, though the rate of decrease is larger for  $\sigma_{obs}$  than  $\hat{\sigma}_{calc}$ . For both, there exists variance at all values other than the highest and lowest.

## 4.5 Response Surface

The goal, of course, is not the interval length  $L_{0.95}(\sigma_{comp}, \hat{\sigma}_{calc})$ , but rather the probability that the interval length is less than some specified value  $K$  given by  $G_K(\sigma_{obs}, \hat{\sigma}_{calc})$ . One possible approach is to calculate this empirically, but the disadvantage to this approach is that such empirical calculation depends on having a large number of  $L_{0.95}(\sigma_{comp}, \hat{\sigma}_{calc})$ . This is unlikely to be the case in real applications. An alternative is to fit a response surface, such as a Gaussian process or a polynomial response surface, and use uncertainty from the surface to determine estimates of  $G_K(\sigma_{obs}, \hat{\sigma}_{calc})$ . As previously discussed in Section 3.1 and shown in figure 5, the response surface shows non-constant variance. Additional complications for this data set are non-normality of error, particularly for small values of  $\sigma_{obs}$  and  $\hat{\sigma}_{calc}$ , and that any intervals must necessarily account for the fact that a confidence interval can not, by definition, be smaller than zero. These issues are mitigated by applying a log transformation to the interval lengths.

For the log transformed interval lengths, a polynomial response surface was fit using a stepwise regression method. The response surface used weighted least squares estimation method to capture the non-constant variance, with weight for a certain  $(\sigma_{obs}, \hat{\sigma}_{calc})$  point proportional to the inverse of the variance of interval lengths for all 100 simulations at each corresponding  $(\sigma_{obs}, \sigma_{comp})$ . Prediction intervals were obtained. The log confidence interval widths, estimated response surface, and 95% prediction intervals are shown below in Figure 6.

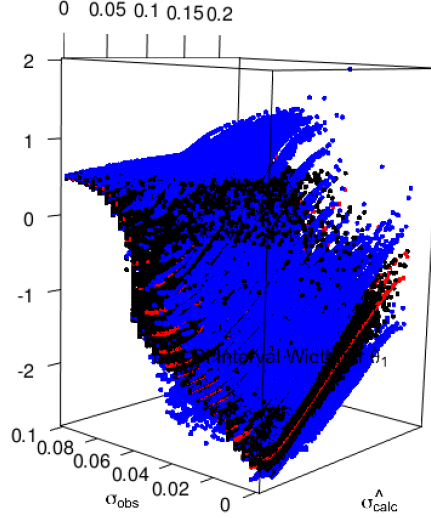


Figure 6: The log interval widths with predicted values given by a polynomial response surface found using stepwise regression. The red points are predicted values and the blue points show upper and lower 95% prediction interval boundaries.

Using the standard error for prediction, calculation of  $G_K(\sigma_{obs}, \hat{\sigma}_{calc})$  is direct for any given  $K$ . Suppose that, for the diffusion equation calibration problem, the experimenter is interested in  $K = -1.386$  for the log response surface, which is equivalent to  $K = 0.25$  on the untransformed interval widths. Given a predicted mean interval width  $\hat{L}$  with estimated standard error of prediction  $se(\hat{L})$ , determined directly from the response surface, the quantity  $G_{-1.386}$  is given by

$$G_{-1.386}(\sigma_{obs}, \hat{\sigma}_{calc}) = \int_{-\infty}^{-1.386} \phi\left(\frac{x - \hat{L}}{se(\hat{L})}\right) dx$$

where  $\phi(\cdot)$  is the standard normal probability density function.

This may be applied to each  $(\sigma_{obs}, \hat{\sigma}_{calc})$  to obtain the function  $G_{-1.386}$ . Lastly, suppose that a  $p = 0.8$  is desired for the resulting log interval width to be less than  $-1.386$ . This area is the region  $B_{0.8, -1.386}$ , and can be determined empirically. A plot of  $G_{-1.386}$  is shown below in Figure 8 with the region  $B_{0.8, -1.386}$  highlighted in red.

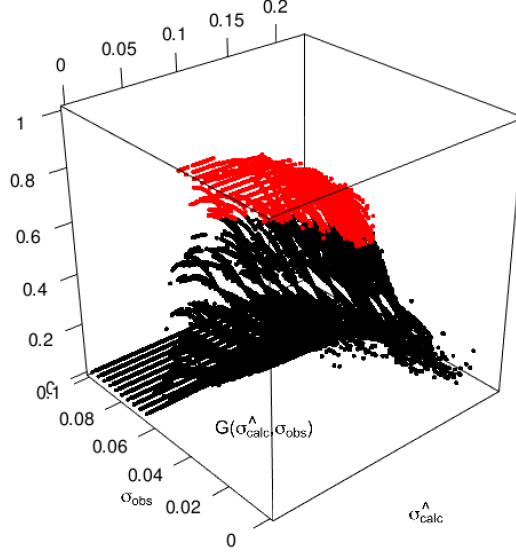


Figure 7: For the response surface in Figure 6 , a plot of  $G_{-1.386}(\sigma_{obs}, \hat{\sigma}_{calc})$ . Points colored red are those above a probability threshold of  $p = 0.8$ , indicating an 80% or higher probability that the resulting interval width obtained from these levels of observational and computational error will be below  $e^{-1.386} = 0.25$ . This region is  $B_{0.8, -1.386}$ .

## 4.6 Costs and Boundaries

Lastly, suppose that the cost of using a particular  $(\sigma_{obs}, \hat{\sigma}_{calc})$  pair is given approximately by the function

$$C(\sigma_{obs}, \hat{\sigma}_{calc}) = |\log(\sigma_{obs})| + 2|\log(\hat{\sigma}_{calc})| - |\log(\sigma_{obs} + \hat{\sigma}_{calc})| \quad (10)$$

Then plugging  $B_{0.8, -1.386}$  into Equation (10) to obtain  $C(B_{0.8, -1.386})$ , the cost surface for the ideal region is obtained, and shown below in Figure 8.

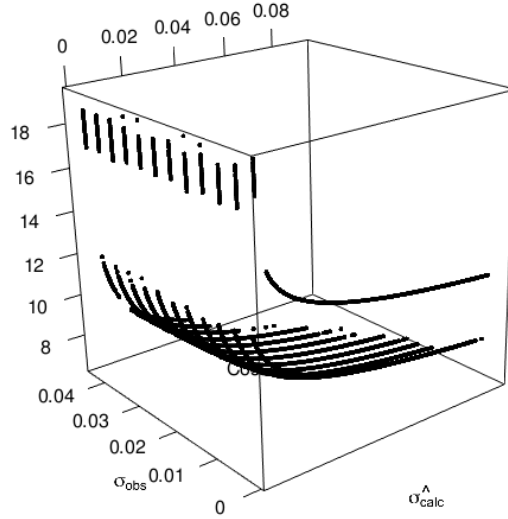


Figure 8: For the  $B_{0.2,-1.386}$  region highlighted in Figure 8 , a plot of the cost of each  $(\sigma_{obs}, \hat{\sigma}_{calc})$  is shown using Equation 10.

From this surface, the cost at each point on the boundary of the region, or any point within, may be obtained. Ideally, the goal would be to choose value of  $\sigma_{obs}$  and  $\hat{\sigma}_{calc}$  that minimize the cost. For the cost function in Equation (10) and the diffusion equation data, the cost is minimized by using  $\sigma_{obs} = 0.0297$  and  $\hat{\sigma}_{calc} = 0.0818129$ , corresponding to a  $\sigma_{comp} = 0.0334$ .

## 5 Conclusion and Future Work

This paper has described a reasonable framework for modeling both computational and observational uncertainty in computer experiments. The goal . To that extent, a framework, including notation, has been developed for incorporating and comparing both observational and computational uncertainty into a coherent framework, and for comparing the effects of both on the resulting quantities of interest. This framework has been applied to an example using the Duffing equation, and is able to accurately identify a potential set of observational and computational uncertainty that optimizes a cost function.

The diffusion equation example of Section 4 implemented the framework of Section 3 only after a time-intensive large number of simulations had been performed in order to easily estimate the boundary for the region  $B_p$ . To be used in

practice, this method must be expanded to encompass a stepwise search for the boundary based likely only off an initial sample within the space of  $\sigma_{obs}$  and  $\hat{\sigma}_{calc}$ . The key difficulty to this search is the non-constant variance. In the polynomial response surface in Figure 6, a weighted least squares estimation technique was used with weights determined using the large number of simulations. In a real search, this resource is unavailable, and so any parameters or properties of the non-constant variance must be estimated as the search is in progress. This poses a formidable challenge, but there exists information. The form of the heteroskedasticity is likely to remain constant, as variance will tend to be smallest in areas of simultaneous large and simultaneous small values of  $\sigma_{obs}$  and  $\hat{\sigma}_{calc}$ , and larger where one is small and the other large. The error is likely to be skewed towards larger values, as confidence interval lengths are bounded below by zero. Finally, the variance will likely be related to the rate of change of the response surface. If this problem can be solved, the iterative procedure of Section 3.1 may be used to approximate the costs and benefits of lowering computational or observational uncertainty before an experiment is conducted.

## References

- Baker, A. H., Hammerling, D. M., Levy, M. N., Xu, H., Dennis, J. M., Enton, B., Edwards, J., Hannay, C., Mickelson, S. A., Neale, R., Nychka, D., Shollenberger, J., Tribbia, J., Vertenstein, M., and Williamson, D. (2015), “A new ensemble-based consistence test for the Community Earth System Model (pyCECT v1.0),” *Geoscientific Model Development*, 8, 2829–2840.
- Bect, J., Ginsbourger, D., Li, L., Picheny, V., and Vazquez, E. (2012), “Sequential design of computer experiments for the estimation of a probability of failure,” *Statistics and Computing*, 22(3), 773 – 793.
- Cavin, R. K., Lugli, P., and Zhirnov, V. V. (2012), “Science and Engineering Beyond Moore’s Law,” *Proceedings of the IEEE*, 100(Special Centennial Issue), 1720–1749.
- Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., and Richet, Y. (2014), “Fast Parallel Kriging-Based Stepwise Uncertainty Reduction With Application to the Identification of an Excursion Set,” *Technometrics*, 56(4), 455 – 465.
- Fehlberg, E. (1969), Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems,, Technical Report 315, NASA.
- Gardner, C. (2018).  
**URL:** <https://math.la.asu.edu/gardner/522.html>
- Higdon, D., Kennedy, M., Cavendish, J., Cafo, J., and Ryne, R. (2004), “Combining Field Data and Computer Simulations for Calibration and Combining Field Data and Computer Simulations for Calibration and Prediction,” *SIAM Journal on Scientific Computing*, 26, 448–466.
- Kennedy, M., and O’Hagan, A. (2001), “Bayesian Calibration of Computer Models,” *Journal of the Royal Statistical Society B*, 63(3), 425 – 464.
- Marques, A. N., Lam, R. R., and Willcox, K. E. (2018), “Contour location via entropy reduction leveraging multiple information sources,” , arXiv:1805.07489 [stat.ML].
- Milroy, D. J., Baker, A. H., Hammerling, D. M., Dennis, J. M., Mickelson, S. A., and Jessup, E. R. (2016), “Towards characterizing the variability of statistically consistent Community Earth System Model simulations,” *Procedia Computer Science*, 80, 1589 – 1600.

- Ranjan, P., Bingham, D., and Michailidis, G. (2008), “Sequential Experiment Design for Contour Estimation From Complex Computer Codes,” *Technometrics*, 50(4), 527–541.
- Stensrud, D. J., Bao, J.-W., and Warner, T. T. (2000), “Using Initial Condition and Model Physics Perturbations in Short-Range Ensemble Simulations of Mesoscale Convective Systems,” *Monthly Weather Review*, 128(7), 2077–2107.
- Victor, P., Ginsbourger, D., Roustant, O., Haftka, R. T., and Kim, N.-H. (2010), “Adaptive Designs of Experiments for Accurate Approximation of a Target Region,” *Journal of Mechanical Design*, 132(7).